Detection of Misbehavior Nodes Using Voronoi Distance Regions and Pivot Distance Regions in Cognitive Networks

Y.Keerthi*1, C.Rajendra*2

M.Tech (CSE) Student, Dept of CSE, ASCET, Gudur, D.t: Nellore, A.P, India

Professor, Dept of CSE, ASCET, Gudur, D.t: Nellore, A.P, India

ABSTRACT:

Data storage has become an important issue in sensor networks as a large amount of collected data need to be archived for future information retrieval. This paper introduces storage nodes to store the data collected from the sensors in their proximities. The storage nodes alleviate the heavy load of transmitting all the data to a central place for archiving and reduce the communication cost induced by the network query. This paper considers the storage node placement problem aiming to minimize the total energy cost for gathering data to the storage nodes and replying queries.

We examine deterministic placement of storage nodes and present optimal algorithms based on dynamic programming. Further, we give stochastic analysis for random deployment and conduct simulation evaluation for both deterministic and random placements of storage nodes. Low power, high performance sensor designs are of particular importance nowadays considering the plethora of emerging applications in a wide range of fields. Applications such as audio capture, FFT (Fast Fourier Transform), image sensing, motion detection, feature extraction and cestrum calculations generate raw data in much larger quantity than more mundane applications as temperature and humidity sensing. In-situ processing of data is an effective way to reduce the amount of information needed to be transmitted over the wireless links. The Co-S platform which we develop addresses this very crucial aspect of sensor network design. The Co-S integrated with a System on Chip (SoC) based host platform and a gigabyte scale energy efficient data storage system, simultaneously satisfies the constraints of low power consumption and a small form factor.

I. INTRODUCTION

Many sensor network applications that are related to pervasive computing, e.g., monitoring learning behavior of the children, senior care system, environment sensing, etc., generate a large amount of data continuously over a long period of time. Often, the large volumes of data have to be stored somewhere for future retrieval and data analysis. One of the biggest challenges in these applications is how to store and search the collected data. The collected data can either be stored in the network sensors, or transmitted to the sink. Several problems arise when data are stored in sensors. First, a sensor is equipped with only limited memory or storage space, which prohibits the storage of a large amount of data accumulated for months or years. Second, since sensors are battery operated, the stored data will be lost after the sensors are depleted of power. Third, searching for the data of interest in a widely scattered network field is a hard problem. The communication generated in a network-wide search will be prohibitive.

Alternatively, data can be transmitted back to the sink and stored there for future retrieval. This scheme is ideal since data are stored in a central place for permanent access. However, the sensor network's per-node communication capability (defined as the number of packets a sensor can transmit to the sink per time unit) is very limited. A large amount of data cannot be transmitted from the sensor network to the sink efficiently. Furthermore, the data communication from the sensors to the sink may take long routes consuming much energy and depleting of the sensor battery power quickly. In particular, the sensors around the sink are generally highly used and exhausted easily, thus the network may be partitioned rapidly. It is possible that, with marginal increase in cost, some special nodes with much larger permanent storage (e.g., flash memory) and more battery power can be deployed in sensor networks. These nodes back up the data for nearby sensors and reply the queries. The data accumulated on each storage node can be transported periodically to a data warehouse by robots or traversing vehicles using physical mobility as Data Mule. Since the storage nodes only collect data from the sensors in their proximity and the data are transmitted through physical transportation instead of hop-by-hop relay of other sensor nodes, the problem of limited storage, communication capacity, and battery power is ameliorated. Placing storage nodes is related to the sensor network applications. We believe query is the most important application for sensor networks since in essence sensor network is about providing information of the environment to the end users. A user query may take various forms; for example, a user query may be how many nodes detect vehicle traversing events, the average temperature of the sensing field, etc.

In this scenario, each sensor, in addition to sensing, is also involved in routing data for two network services: the raw data transmission to storage nodes and the transmission for query diffusion and query reply. Two extremes, as mentioned earlier, would be transmitting all the data to the sink or storing them on each sensor node locally. On one hand, data solely stored in the sink is beneficial to the query reply incurring no transmission cost, but the data accumulation to the sink is very costly. On the other hand, storing data locally incurs zero cost for data accumulation, whereas the query cost becomes large because a query has to be diffused to the whole network and each sensor has to respond to the query by transmitting data to the sink. The storage nodes not only provide permanent storage as described previously, but also serve as a buffer between the sink and the sensor nodes.

II.RELATED WORK

Wireless Sensor Networks

Wireless Sensor Networks (WSNs) are deployments of low power devices which are equipped with a multitude of features such as a processor, a radio, flash memory, and several environmental monitoring sensors. WSNs are expected to help researchers in monitoring environmental conditions at a high fidelity. Deployments of WSNs have already emerged in environmental and habitant monitoring, seismic and structural monitoring, factory and process automation and a large array of other applications. These nodes may be used for sensing and reporting events as and when they occur, or sensing and logging on a regular time schedule.

WSNs coagulate data from the sensing nodes throughout the network at the sink. Sensor platforms designed to live up to the expectations of the deployment specifications need not only to be power efficient [28] but also must incorporate the ability to reduce the amount of information transferred via wireless links over the network in order to avoid depleting the complete setup of precious battery resources. Popular embedded sensor network architectures like Mica, Wisenet, Rene, Telos and iBadge, employing power aware computing methodologies have been deployed successfully for a wide range of applications such as temperature, pressure, luminosity measurements.

Applications aimed at this domain are no longer restricted to the mundane, temperature and humidity sensing. In fact researchers working on sensor networks strive to conjure up diverse applications such as intrusion detection systems, voice based reporting systems, ecological monitoring structural monitoring and many more. Applications such as these demand an ever greater level of processing capability from sensor nodes which are deployed in the field. Cepstrum calculation for resonance analysis, FFT, harmonic filtering and extraction, image masking and processing all require a significant level of processing capability to be demonstrated by the Micro Controller Unit (MCU) on the small power efficient nodes. Current implementations of such platforms, mostly utilize 8-bit MCU's which are unsuitable for the job. They simply do not possess enough processing capability to satisfy the compute intensive applications which are now being hived onto this domain. Through extensive experimentation and benchmarking we have developed a sensor co-processing (Co-S) architecture integrated with an System on Chip based, RISE host platform for higher performance sensing needs which simultaneously satisfies the constraints of low power consumption, high computational capability, high capacity onboard storage and a small form facto

III.MOTIVATION

Our work is motivated by the requirements of the Bio-Complexity and the James Reserve Projects at the Center of Conservation Biology (CCB) at UC Riverside. CCB is working towards the conservation and restoration of species and ecosystems by collecting and evaluating scientific information. The bio-complexity project is designed to develop the kinds of instruments that can monitor the soil environment directly in environments where factors like high humidity and precipitation will be a challenge for the sensors, rather than in laboratory recreations. One of the goals is to improve understanding of the spatial and temporal processes that control soil carbon sequestration in a tropical seasonal forest and the role of soil micro-organisms. The objectives in particular are to study soil carbon in a fire chronosequence to evaluate on ecological restoration experiment in terms of carbon and to integrate spatially and temporally the information from the sensor arrays with ecosystem scale measurements (e.g. root biomass, litter, soil carbon). Additionally voice signature based recognition mechanisms need to be implemented on the sensor platforms for habitat monitoring, enabling identification of species of birds using certain distinguishing features possible with frequency domain analysis of their native call patterns (songs). Our objectives from the ground up are to reduce power consumption, maintain software compatibility vis-à-vis TinyOS and simultaneously broaden the spectrum of applications of compact sensor systems. In keeping with our goals, our host architecture employs a monolithic SoC device viz. ChipCon CC1010, which includes a power optimized 8051 core, radio, 3 ADCs (Analog to Digital Converter), 2KB SRAM (Static Random Access Memory), 32 KB on-chip flash, 2 UARTs (Universal Asynchronous Receiver Transmitter), SPI (Serial Peripheral Interface) bus, all onto a single SoC architecture running TinyOS networking stack and an interface layer with the Co-S. On one hand this simplifies hardware design due to integration of all the components onto a single chip, eliminating a complex interface, while on the other hand overall system power consumption is reduced due to tightly integrated peripherals on the chip. Since off chip peripherals entail individual *Printed Circuit Board (PCB)* area, voltage drop through the longer PCB traces, leakage and quiescent operation currents, our single chip host architecture is effectively power advantaged over discrete systems. Our architecture differs from existing platforms not only in terms of its computing power, flexibility, level of on-chip component integration but also, and quite significantly in the amount of on-board storage memory that it can provide, and an added software paradigm of "Sense and Store" which manages humongous amounts of raw data from the sensing hardware in-situ, before transmitting relevant parts of it efficiently to the base station. The extent of potential savings in energy brought about by employing the "Sense and Store" paradigm versus the "Sense and Send" is illustrated in Figure 1. Our "Sense and Store" paradigm stores the data onto the onboard memory, instead of naively passing on each and every piece of raw data through the hierarchical structure of a sensor network. This new approach has been demonstrated to be a substantial improvement over the Sense and Send architecture.

The moot question that needs to be answered now is whether the integration of components on chip, along with employing latest hardware and software paradigm is truly beneficial, and how could the underlying benefits be quantified. Thus to answer this question we have quantitatively compared our platform with various other sensing platforms including the crossbow MICA [4]

IV. Problem Formulation

In this paper, we consider an application in which sensor networks provide real-time data services to users. A sensor network is given with one sensor identified as the sink (or base station) and each sensor generating (or collecting) data from its environment. Users specify the data they need by submitting queries to the sink and they are usually interested in the latest readings generated by the sensors1. There are two types of sensors (or nodes) in this hybrid network, defined as follows.

• *Storage nodes*: This type of nodes store all the data it has received from other nodes or generated by themselves.

The sink only sends queries to storage nodes. According to the query description, storage nodes obtain the results needed from the raw data they are holding and then send these results back to the sink. The sink itself is considered as a storage node.

• *Forwarding nodes*: Each forwarding node is associated with a storage node. A forwarding node always forwards the data generated by itself to the associated storage node. Since forwarding nodes are not aware of queries, the forwarding operation is independent of queries and there is no data processing at these nodes.

V. MODELS

A. System Model

We consider a sensor network consisting of storage nodes and regular sensors. The basic query/response model is illustrated in Fig. 1.

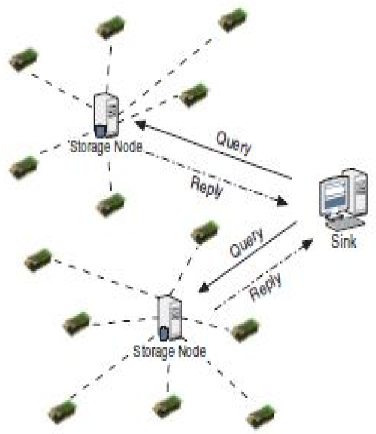

Fig. 1. Two-tiered System Model (with two storage nodes)

We assume that every sensor generates environmental data values in a fixed rate and periodically submits the collected data to the closest storage node. For example, sensors monitor temperature every ten seconds and submit the data to storage nodes every one minute. Thus, each submission contains six temperature readings. We define an *epoch*, as the interval time between two submissions (one minute in the above example). Assume all sensors are synchronized so that they have agreement on the beginning and end of an epoch. The data messages from sensor *si* contain the following information:

$si \rightarrow$ Storage Node : $i, t, \{data1, data2, \ldots\}$,

where i is the sensor ID and t is the current value of the epoch counter. Data query from a user is directed to the storage nodes through the sink. In this paper, we consider range queries in the following format $RangeQuery = \{t, [a, b]\}$, where t is the time slot the user is interested in and $[a, b]$ is the specified data value range. For easy exposition, we only consider one dimensional data in this paper. In some applications, sensors may generate data with multiple attributes, which yield more complex range query. Our approach, however, can be easily extended to the query with multiple data types.

B. Adversary Model and Security Goals

We assume that the adversary tries to launch the following two attacks. First, the adversary wants to obtain the sensitive data information from the sensor network, which violates *data privacy*. Leaking valuable data is a critical threat in many applications. The second attack is to breach *data fidelity*. For a user's query, the adversary tries to reply with wrong information and makes the user accept it. We consider that both storage nodes and regular sensors might be compromised in a hostile environment. In the rest of this subsection, we discuss the impacts of the compromised storage nodes and regular sensors, and propose our corresponding security goals.

*1) **Compromised Storage Nodes**:* Our major focus is on the compromised storage nodes. Since storage nodes host a lot of data collected from other regular sensors, compromising storage nodes will cause great damages to the system. First, once compromising a storage node, the adversary easily obtains the privacy-sensitive data stored on the storage node. Second, the compromised storage nodes can help the adversary launch the *data fidelity* attack, because storage nodes are responsible for answering queries from the sink. After receiving a query, the compromised storage nodes may return arbitrary data as the reply. Therefore, this paper has the following security goals for the compromised storage nodes. We aim to protect *data privacy* by designing a storage scheme, such that little information is exposed to storage nodes while fulfilling data queries. *Data fidelity* attack, however, is hard to prevent. Our countermeasure is an approach to enabling the sink to detect and reject the false reply.

*2) **Compromised Sensors**:* If one regular sensor is compromised, the following readings of the sensor will be exposed and the adversary may send forged data to storage nodes.

Unfortunately, it is hard to prevent the data privacy attack and data fidelity attack in this scenario. However, the data from an individual sensor is minor in the whole network. Unless the adversary compromises a lot of regular sensors, this kind of attack has a very limited impact.

VI.CONCLUSION

We have developed a highly integrated sensing platform utilizing SoCs thus, resulting in an overall simplified design. Our design integrates a powerful co-processing SoC that effectively broadens the spectrum of applications suitable for small and low powered sensing systems. We have also maintained complete software / network compatibility by using the TinyOS on the host platform. Our research effort has been able to prove conclusively the efficacy of the "Sense and Store" paradigm over the "Sense and Send" methodology being employed by most conventional sensor deployments in today's scenario. We have demonstrated a performance improvement of seventy times in terms of power consumption following the "Sense and Store" approach. Furthermore, our Co-S architecture leads to a significant improvement in terms of computational ability, twenty four times more than the conventional low power sensors in use today. Co-S platform coupled with the on-board gigabyte scale data storage can now enable computation intensive applications such as FIR, FFT, Harmonic extraction, voice signature mapping and a plethora of similar demanding applications to be run on small and low power sensor systems. Large capacity SD-Cards hooked onto our platform allows accumulation of gigabyte scale sensed data onto the sensor itself. And coupled with the "Sense and Store" approach, is a win-win situation for sensor networks not only in terms of power savings but also in terms of the amount of data available for post-mortem purposes.

Additionally our design sports the novel ability to maintain network connectivity in the face of power depletion by terminating all sensing operations, by shutting down the sensing module, thus enhancing the longevity of the network.

VII. REFERENCES

[1] F. Baccelli, M. Klein, M. Lebourges, and S. Zuyev. Stochastic geometry and architecture of communication networks. J. Telecommunication Systems, 7:209–227, 2013.

[2] F. Baccelli and S. Zuyev. Poisson-voronoi spanning trees with applications to the optimization of communication networks. Operations Research, 47(4):619–631, 2009.

[3] S. J. Baek, G. de Veciana, and X. Su. Minimizing energy consumption in large-scale sensor networks through distributed data compression and hierarchical aggregation. IEEE JSAC Special Issue on Fundamental performance limits of wireless sensor networks, 22(6):1130–1140, August 2004.

[4] S. Bhattacharya, H. Kim, S. Prabh, and T. Abdelzaher. Energy-conserving data placement and asynchronous multicast in wireless sensor networks. In Proceedings of the 1st international conference on Mobile systems, applications and services, pages 173–185, New York, NY, USA, 2003. ACM Press.

[5] A. Demers, J. Gehrke, R. Rajaraman, N. Trigoni, and Y. Yao. The cougar project: a work-in-progress report. SIGMOD Rec., 32(4):53–59, 2003.

[6] P. Desnoyers, D. Ganesan, H. Li, M. Li, and P. Shenoy. PRESTO: A predictive storage architecture for sensor networks. In Tenth Workshop on Hot Topics in Operating System, 2005.

[7] E. J. Duarte-Melo and M. Liu. Data-gathering wireless sensor networks: organization and capacity. Computer.

[8] D. Ganesan, D. Estrin, and J. Heidemann. Dimensions: why do we need a new data handling architecture for sensor networks? SIGCOMM Comput. Commun. Rev., 33(1):143–148, 2003.

[9] D. Ganesan, B. Greenstein, D. Perelyubskiy, D. Estrin, and J. Heidemann. An evaluation of multi-resolution storage for sensor networks. In Proceedings of the 1st international conference on Embedded networked sensor systems, pages 89–102, New York, NY, USA, 2003. ACM Press.

[10] P. Gupta and P. R. Kumar. The capacity of wireless networks. IEEE Transactions on Information Theory, IT-46(2):388–404, March 2000.